

Feature 39107

Version 3 des flux

Spécification fonctionnelle détaillée



VERSIONING

Date	Auteur	Sujet
06/03/2025	Irène BURLET	Création du document
16/06/2025	Irène BURLET	Suppression de la mention de la non prise en compte des flux v1 par ikos wrapper + ajout de 3 nouveaux champs à créer côté api (fin du tableau)
18/06/2025	Irène BURLET	Ajout d'un traitement de l'existant + Modifications diverses sur la partie solution technique - Candidats

VERSIONING.....1

LISTE DES TICKETS DE LA FEATURE.....2

DESCRIPTION FONCTIONNELLE.....3

SCÉNARIOS.....4

AUTORISATIONS.....6

TRAITEMENT DE L'EXISTANT.....6

LIVRAISON ET DÉPENDANCES.....6

SOLUTION TECHNIQUE.....6

CHAMPS DU MODEL HOUSING_OFFERS.....7

FLUX 1.....11

DESCRIPTION.....12

LOGIQUE DE LA MÉTHODE.....13

RETOURS.....13

FLUX 3 - CANDIDATS.....14

FLUX 4 - CANDIDATS.....15

APPEL API.....15

LOGIQUE DE LA MÉTHODE.....15

RETOURS.....16

CODE MOTIF ÉVÉNEMENT.....16

PRISE EN COMPTE DU CODE ÉVÉNEMENT.....17

GESTION DE L'ÉVÉNEMENT DESCAL04 (REFUS ATTRIBUTION).....17

GESTION DE L'ÉVÉNEMENT ANN RAD (ANNULATION DE RADIATION).....18

GESTION DE L'ÉVÉNEMENT NPR CAL (NON PRÉSENTATION EN CAL).....19

GESTION DE L'ÉVÉNEMENT DESCAL03 (AJOUT DU 18/06).....19

FLUX 5.....	19
LOGIQUE DU FLUX 5.....	20
FLUX 5 UNIFIÉ.....	20
CHANGEMENTS POUR LA CRÉATION DU XML.....	20
RÉFÉRENTIEL DE TYPOLOGIES DE LOGEMENT.....	20
MODIFICATION DE L’IHM COLLABORATEUR ALS.....	21
AJOUT DE TESTS POUR LE FLUX 5.....	21
 <u>ANNEXES.....</u>	 <u>21</u>
 FLUX 5 UNIFIÉ.....	 22

LISTE DES TICKETS DE LA FEATURE

Numéro	Titre	Partenaires
39364	Gestion des nouvelles zones FLux 1	Be'ys / Sopra
39366	Gestion des nouvelles zones Flux 3 candidat	Sopra ?
39365	Gestion du FLux 2	Be'ys
39367	Gestion du Flux Evènement (Flux 4)	Be'ys / Sopra

DESCRIPTION FONCTIONNELLE

Cette documentation est majoritairement basée sur la documentation be'ys "FLXv3 - Flux V3 - 39107 - documentation - fra v1.0.docx".

L'objectif de la feature est de faire évoluer :

- Le flux 1 - Patrimoine (allant du bailleur à ikos wrapper puis la base al'in) : pour ajouter/modifier/supprimer les balises du xml décrivant les offres envoyées par le bailleur
- Le flux 4 - Evénements (allant du bailleur à ikos wrapper puis la base al'in) pour permettre aux bailleurs d'envoyer 1 nouvel événement et 2 motifs d'événements
- Les tableaux de suivi (pas d'impact côté Al'in)

Ces évolutions seront portées par une nouvelle version 3 des flux bailleurs, nouvelle version qui doit permettre lors de sa mise en exploitation une rétrocompatibilité avec la version 2 des flux bailleurs actuellement en production.

Parallèlement, les flux v2 font l'objet d'une mise à jour les passant de v2.1.9 à v2.2.0.

Scénarios

1. Flux 1 – Patrimoine

Le flux 1 consiste à :

- Envoyer sous format xml des offres, xml générés par les bailleurs, dans un numéro de version donné
- Les xml sont traités par ikos wrapper quotidiennement afin d'alimenter la base Al'in
- Ikos wrapper est donc en charge de retranscrire les informations transmises via balises xml en valeurs à envoyer via api json pour les champs correspondants en base Al'in
- Si un bailleur envoie un premier xml dans une version supérieure à la précédente, alors on considère que le bailleur ne doit plus envoyer que des flux dans cette version. Par exemple si version_flux = 3 alors le bailleur ne pourra envoyer que des flux en v3 par la suite

Nous devons distinguer, aussi bien côté Ikos Wrapper que côté Al'in, la gestion du traitement des données en fonction de la version du flux.

Actuellement si un bailleur envoie un fichier sans bien, aucun traitement n'est effectué et Ikos Wrapper passe au fichier suivant. L'attendu est désormais que les contrôles autres que l'intégration d'offre soient effectués à la réception d'un flux 1 sans offres. Ce changement n'a pas d'impact côté Al'in.

2. Flux 3 – Candidats

Dans le cadre du flux 3 qui concerne les candidatures (appel d'Al'in Gestion vers Al'in), une nouvelle balise **identifiant_candidat_ALS** a été mise en place mais elle ne concerne que les cas où aucun nud n'est passé, dans le cadre du logement temporaire. Côté Al'in nous ne sommes pas impacté par ce changement, un nud devant toujours nous être fourni. Si elle nous est envoyée, la nouvelle balise sera ignorée et nous devons renvoyer une erreur si le nud transmis par AL'in Gestion n'est pas valide et aucune action ne sera faite côté Al'in.

3. Flux 4 - Événements bailleur

La même balise **identifiant_candidat_ALS** que pour le point ci-dessus a été mise en place dans le xml de l'événement. De même que ci-dessus, ce changement n'impacte pas Al'in.

Par contre il est prévu que :

- Un nouvel événement nous soit transmis (DESCAL04)
- Deux nouveaux motifs (ANNRAD + NPRCAL)

Après étude, nous devons adapter la façon dont on gère la “traduction” entre statut Al'in et événement bailleur, afin de pouvoir facilement les interpréter et les transmettre à Al'in Gestion. En effet, nous avons avant une équivalence “statut Al'in” = “code événement”. Par exemple : REFUSEE EN CAL = DESCAL03, . Mais dans le cadre de cette feature, il est demandé qu'une candidature :

- REFUSEE EN CAL puisse avoir le code événement DESCAL04, DESCAL03 ou NPRCAL
- TRANSMIS BAILLEUR puisse avoir le code événement ANNRAD

Donc nous allons créer un nouveau champ pour avoir plus de finesse sur le type d'événement reçu.

Nous devons également mettre à jour les informations sur les objets liés à la candidature comme ceci (détails techniques dans la partie “Solution technique”) :

- DESCAL04 et NPRCAL :
 - o Changements sur la demande pour recalculer l'entity_status et les références liées à la candidature
- ANNRAD :
 - o Changement sur l'offre pour la modification du statut (offer_status = EN COURS D'ATTRIBUTION et lessor_status = EN ATTRIBUTION)
 - o Changements sur la demande pour recalculer le request_status, l'entity_status, le is_active, les champs liés à l'offre attribuée sont vidés, les liaisons avec la candidature sont remises

4. Flux 5 – Envoi des offres à Al'in Gestion par Al'in

Il existe un automatisme côté Al'in pour envoyer à Al'in Gestion les offres mises à disposition, aussi bien au moment de leur mise à disposition qu'en cas de modification ensuite. Cet automatisme concerne aussi bien les offres créées via flux que via front. Les conditions d'envoi de l'offre sont détaillées en annexes. Les conditions d'envoi vont évoluer en fonction des modifications apportés sur la liste des champs à inclure dans le xml.

La présence de différentes versions de flux implique que les offres aient des variations sur les champs renseignés en base, il faut donc prévoir de désormais envoyer à Al'in Gestion dans le cadre du flux 5 le numéro de version associé à l'offre, dans la balise "version_flux".

5. Création d'offre via front bailleur

Afin de ne pas impacter les offres créées depuis le front Al'in, si aucune version de flux n'est renseignée alors il faut enregistrer la valeur "2".

Autorisations

Dans le cadre de cette feature, il n'est pas prévu que des modifications fonctionnelles impliquant l'accès aux offres soient faites. Les appels de flux sont faits entre outils non accessibles via interface, avec des comptes administrateurs.

Traitement de l'existant

~~Il n'est pas nécessaire de prévoir de script afin de reprendre les données existantes. Les anciennes offres reçues avant livraison seront dans leur version de flux actuel, et les nouvelles offres créées après livraison seront en version 2 ou 3. Les champs communs à ces 2 versions étant prévus d'être maintenus dans leur fonctionnement actuel.~~

~~Pour l'envoi des offres vers Al'in Gestion, les offres créées avant livraison (donc sans indication sur le numéro de version du flux dont elles sont issues), seront traitées comme étant au format v2.~~

Changement du 18/06 :

Il est nécessaire de retravailler les données existantes pour prendre en compte l'évolution concernant la typologie de logement. En effet il faut recalculer l'éligibilité au niveau des demandes pour prendre en compte le fait que les demandes éligibles à des offres T6P puissent voir les offres ayant T9P. Il n'est pas souhaité que les offres soient traitées cependant, celles-ci ayant une durée de vie courte.

Livraison et dépendances

Cette feature nécessite la livraison de 3 outils : Ikos Wrapper, Al'in et Al'in Gestion.

Après livraison, il faut prévoir de configurer le paramétrage hermes_config, champ "motive_label" afin d'inclure les nouveaux codes motifs et libellés associés pour les événements bailleurs du flux 4.

SOLUTION TECHNIQUE

Nous allons décrire dans cette spécification les principaux appels api existants concernés par la feature et les modifications à y apporter, ainsi que les modifications à prévoir sur les champs de l'offre.

Champs du model housing_offers

Les offres sont décrites dans la base Al'in via le modèle housing_offer. Ce modèle n'est pas

1. Champs supprimés

Champ Al'in	Balise XML
first_quartile	positionnement_quartile_1
other_quartiles	positionnement_quartiles_autres
priority_public	public_prioritaire

Ces champs étant supprimés, ils ne pourront plus être cherchés, modifiés ni transmis à Al'in Gestion via flux 5.

2. Champs modifiés

a. Champs devenus facultatifs

Champ Al'in	Balise XML
furnished_lodging	meuble
lessor_phone_number	telephone_a_afficher

b. Autres modifications

Champ Al'in	Balise XML	Nature du changement
parking_type	(V2) <stationnements_residence>	Pas de changement côté Al'in, le champ reste une chaîne de

	(facultative) <stationnements_logement> (fac.) <stationnement_logement_nature > (1,*) ... </> </>	caractère renseignée via api, avec des valeurs séparées par un caractère spécial
parking_type	(V3) stationnement_logement_nature	Côté Ikos Wrapper : valeurs (codes) de stationnement_logement_nature à compiler sous forme de string avec des pipes pour séparateur (pas de changement côté Al'in)
designation_date_end	date_fin_designation	Il faut : - autoriser la modification du champ designation_date_end sur une offre en V3 - conserver le calcul dynamique (non modifiable via endpoint) pour les offres en v2
insee_code	ville_insee	Ikos Wrapper : valeur de ville_insee à enregistrer dans le champ existant du modèle offre : insee_code. En V3, contrairement aux V1/V2, ne pas lancer de recherche du code INSEE. Pour la recherche de latitude et longitude, si elles ne sont pas fournies, on utilise aussi la valeur passée dans ville_insee par le bailleur. Cette valeur ne doit être modifiable que via le formulaire bailleur d'édition d'offre.
effort_rate	taux_effort	
accept_low_occupation	sous_occupation	
lessor_contact_name lessor_email	contact_a_afficher email_a_afficher	si les champs sont TOUS LES TROIS vides, dupliquer dans ces champs les informations du « contact

lessor_phone_number	telephone_a_afficher	commercial » comme suit : contact_a_afficher = "commercial_contact{lastname}" + " " + commercial_contact{firstname}" email_a_afficher = "commercial_contact{email}" telephone_a_afficher = "commercial_contact{phone_number}"
----------------------------	----------------------	---

De manière générale, nous devons également revoir notre façon de gérer les booléens car ceux-ci sont actuellement de réels booléens (valeur true/false), alors qu'il faut pouvoir stocker 3 valeurs : true / false / nil.

Les champs concernés sont : **condo** (copropriete), **qpv** (QPV) (sans doute géré dans une autre évolution, celle des quartiles), **has_garden** (jardin), **has_elevator** (ascenseur), **duplex** (duplex), **has_guardian** (gardien), **green_area** (espace_vert), **already_inhabited** (deja_loue), **communicate_exact_address** (communiquer_adresse_exacte), **furnished_lodging** (meuble), **virtual_visit** (visite_virtuelle), **has_cellar** (cave)

Nous pouvons envisager de maintenir le fonctionnement actuel pour ces champs (ils restent en booléens simples, à l'exception de QPV), mais de faire en sorte d'avoir le bon format pour les nouveaux champs de type oui/non.

3. Champs ajoutés

Champ Al'in	Balise XML	Description
dataflow_version	version_flux	Valeur par défaut : 2 Numéro de version du flux xml contenant l'offre
owner_siren	SIREN_proprietaire	Balise destinée à porter le SIREN du bailleur propriétaire de l'offre
follow_up_rights	nature_droit	Nature des droits de suite
delivery_date	date_livraison	Date à laquelle le bien a été livré
first_proposal	premiere_mise_a_habit	Indique s'il s'agit d'une première mise à

	ation	l'habitation
latest_rehabilitation	date_derniere_rehabilitation	Date de dernière réhabilitation
first_round_of_original_reserving_entity	premier_tour_reservataire_initial	Indique s'il s'agit du premier tour par phase de commercialisation par réservataire (Selon convention Gestion en flux)
address_complements_2	adresse_complement_2	
door_number	numero_porte	
residence_type	residence_type	Type de résidence
energy_expenditure	montant_depenses_energies_estime	
storey_number	nombre_niveaux_appartement	
loggia	loggia	
cellar	cellier	
monthly_rent_max	loyer_mensuel_maximum	
annex_rent_with_charges	loyer_annexe_charges_comprises	
external_charges	montant_charges_externes	
parking_charges_mandatory	loyer_stationnement_cc_obligatoire	
financing_solution	dispositif_financement	
financing_solution_ref	ref_dispositif_financement	
commercial_contact:	contact_commercial_no	Objet avec les clefs lastname, firstname,

{firstname: ..., lastname: ..., email: ..., telephone: ...}	m (lastname) contact_commercial_pr enom (firstname) contact_commercial_e mail (email) contact_commercial_tel ephone (telephone)	email et telephone
application_manager : {firstname: ..., lastname: ..., email: ..., phone_number: ...}	contact_commercial_no m (lastname) contact_commercial_pr enom (firstname) contact_commercial_e mail (email) contact_commercial_tel ephone (phone_number)	Objet avec les clefs lastname, firstname, email et phone_number
dpe_ges_value	dpe_valeur_ges	
dpe_conso_value	dpe_valeur_conso	
energy_expenditure_ max	montant_depenses_ene rgies_estime_max	

Flux 1

Dans le cas d'un appel depuis Ikos Wrapper, la route api appelée est la même en cas de création que de modification, l'outil ne sachant pas si l'offre existe ou non. C'est côté Al'in que la vérification de l'existence de l'offre est faite.

Description

Verbe : POST

Chemin : {host}/api/dmo/housing_offers

Valeurs pour {host} :

- Recette 2 : api-cerfav4-rcc.be-idys.com
- Recette : als-hermes-scoring-rec.be-idys.com
- Préproduction : als-hermes-demo-api.be-idys.com
- Production : api.al-in.fr

Paramètres :

```
{
  "data": {
    "attributes": {
      "external_ref": String, // Obligatoire
      "typology": String, // Obligatoire
      "rent_amount": Float, // Obligatoire
      "surface": Integer, // Obligatoire
      "availability_date": DateTime, // Obligatoire
      "address": String, // Obligatoire
      "postal_code": String, // Obligatoire
      "threshold_type": String, // Obligatoire
      "threshold_subtype": String, // Obligatoire
      "kind": String, // Obligatoire
      "rental_charges": Float, // Obligatoire
      "construction_type": String, // Obligatoire
      "residence_title": String, // Obligatoire
      "lessor_contact_name": String, // Obligatoire
      "dpe_ges": String, // Obligatoire
      "dpe_conso": String, // Obligatoire
      "lessor_siren": String, // Obligatoire
      ....
    }
  }
}
```

```
}
}
```

Autorisations :

Aucune restriction d'autorisation sur cet appel.

Logique de la méthode

On recherche d'abord l'utilisateur concerné par la requête (via le token ou via les paramètres api si l'utilisateur a le rôle admin). On recherche ensuite l'offre à partir des critères suivants :

```
external_ref
lessor_siren
lessor_code
```

Si l'offre n'est pas trouvée ou qu'il faut la recréer (l'offre a un statut de fin de vie), alors on crée l'offre (s'il n'y a pas d'erreur au moment de la sauvegarde).

Sinon :

- On vérifie si l'utilisateur peut modifier l'offre
- On remplit le user_id de l'offre
- On ne tient pas compte des champs lessor_status, offer_status, reserving_business_name, et details sauf s'il est vide
- On vérifie que les champs de pictures respectent les règles de remplissage (soit tous vides, soit on ne peut plus vider les deux)
- On met à jour l'offre

Retours

Retour en succès : La route retournera un code 200

En annexes fichier : reponse-creation-offre.json

Retour en échec si :

- 422 :
 - o Si une des valeurs requises est absente
 - o Si l'offre existe et que l'utilisateur n'a pas les droits d'écriture sur cette offre (code : **not_writers**)
 - o S'il n'y a pas de paramètres dans la requête (code : **no_valid_parameters**)

Format de retour en cas d'erreur :

```
{
  "errors": [
    {
      "status": 422,
```

```
        "source": {
            "pointer": "/data/attributes/availability_date"
        },
        "detail": "can't be blank"
    }
]
}

Ou

{
    "errors": [
        {
            "title": "translation missing: fr.error_messages.not_writers.title",
            "status": 422,
            "code": "translation missing: fr.error_messages. not_writers.code"
        }
    ]
}
```

Flux 3 - Candidats

Pas de changement à apporter

Flux 4 – Candidats

Appel api

Il faut que l'on adapte la route "update_status" utilisée pour nous informer d'un événement (flux 4). C'est cette route qui est utilisée par IKOS WRAPPER et AL'IN Gestion.

Verbe : POST

Chemin : {host}/api/dmo/housing_applications/update_status

Valeurs pour {host} :

- Recette 2 : api-cerfav4-rcc.be-idys.com
- Recette : als-hermes-scoring-rec.be-idys.com
- Préproduction : als-hermes-demo-api.be-idys.com
- Production : api.al-in.fr

Paramètres :

```
{ "data":  
  { "attributes": {  
    "unique_number": "0130724005665G1198",  
    "external_ref": "nad_offre_0502_5",  
    "last_lessor_event": "RADATT",  
    "last_status_update_date": "2025-05-21T00:00:00Z"  
  }  
}
```

Autorisations :

Pas de changement d'autorisation sur cet appel, seuls les utilisateurs Administrateurs peuvent effectuer l'appel.

Logique de la méthode

Le changement prévu consiste à ne plus avoir les statuts plugandwork envoyés mais directement le code événement (par exemple DESCAL03). Ikos wrapper et Al'in Gestion n'auront donc plus besoin de transposer le code en statut Al'in. Pour rappel voici les correspondances :

```

when 'INSCRIPTION CAL'
"INSCAL"
when 'RETENUE EN CAL SOUS RESERVE'
"DESCAL"
when 'RETENUE EN CAL'
"DESCAL01"
when 'RETENUE EN CAL SOUS CONDITIONS'
"DESCAL02"
when 'REFUSEE EN CAL'
"DESCAL03"
when 'DESISTEMENT'
"REFLOG"
when 'VISITE PROPOSEE'
"VISPRO"
when 'VISITE EFFECTUEE'
"VISEFF"
when 'LOGE'
"RADATT"
when 'ABANDON BAILLEUR'
"ABAN"
when 'A L'ETUDE ALS'
"INSTRU"
when 'REFUSEE EN CAL'
"DESCAL04"
when 'REFUSEE EN CAL'
"NPRCAL"
when 'TRANSMIS BAILLEUR'
"ANNRAD"

```

Les 4 derniers sont des ajouts faits dans le cadre de cette feature.

On renvoie une erreur 422 si le code événement envoyé dans `last_lessee_event` n'est pas connu. Si le code est connu, alors on transpose avec le statut Al'in pour vérifier qu'il n'y a pas de régression de statut en fonction de sa valeur (tous les statuts ne sont pas concernés).

Retours

Retour en succès : La route retournera un code 200

Retour en échec si :

- 422 : `last_lessee_event_unknown`

Format de retour en cas d'erreur :

```

{
  "errors": [
    {
      "title": "translation missing: fr.error_messages.not_writers.title",
      "status": 422,

```

```

    "code": "translation missing: fr.error_messages. not_writers.code"
  }
]
}

```

Code motif événement

Il faut par contre prévoir la modification de la configuration après livraison (admin/settings > hermes_config > champ motive_label) pour ajouter de nouvelles valeurs pour la balise motive_code. Il n’y a pas de restriction sur ce que l’on peut stocker dans ce champ, mais si le code n’est pas déclaré en configuration, alors l’api ne renverra pas de valeur en retour dans les attributs de la candidature, le libelle associé au code n’étant pas connu.

Prise en compte du code événement

Jusqu’ici l’événement bailleur (type DESCAL03) était transformé en statut AL’in au niveau de la candidature (ex : « REFUSEE EN CAL »). Ce statut était ensuite retraduit en code événement lors de l’envoi de l’information à AL’in Gestion (le statut REFUSEE EN CAL était alors retraduit en DESCAL03). Ce fonctionnement n’est tenable que s’il y a une correspondance 1-pour-1 entre codes événements et statuts. Pour intégrer les nouveaux codes événements prévus, il nous faudrait donc créer de nouveaux statuts AL’in, ce qui ne nous semble pas une approche pertinente fonctionnellement, ni tenable à long terme.

Nous allons donc mettre en place une solution qui permette d’avoir de bout en bout la référence au code événement. Accessoirement, cela renforcera la robustesse du système.

Côté AL’in api, nous allons créer un nouveau champ “last_lessor_event” au model housing_application. Lors de l’envoi de l’événement à AL’in Gestion, le code événement utilisé est celui stocké dans le champ “last_lessor_event”, plutôt que d’être reconstruit. Lors d’un changement de statut de candidature n’émanant pas d’un **événement bailleur** (pas de valeur passée pour “last_lessor_event”), le champ “last_lessor_event” passe à null.

Ce nouveau fonctionnement s’appliquera aussi aux événements déjà actuellement transmis. Voici pour rappel la liste des équivalences de statuts :

```

when 'INSCRIPTION CAL'
"INSCAL"
when 'RETENUE EN CAL SOUS RESERVE'
"DESCAL"
when 'RETENUE EN CAL'
"DESCAL01"
when 'RETENUE EN CAL SOUS CONDITIONS'
"DESCAL02"
when 'REFUSEE EN CAL'
"DESCAL03"
when 'DESISTEMENT'
"REFLOG"
when 'VISITE PROPOSEE'
"VISPRO"
when 'VISITE EFFECTUEE'
"VISEFF"

```

```
when 'LOGE'  
"RADATT"  
when 'ABANDON BAILLEUR'  
"ABAN"
```

Gestion de l'événement DESCAL04 (refus attribution)

Lors d'un refus d'attribution par le bailleur, ce dernier envoie un événement DESCAL04 sur la candidature pour notifier AL'in de sa décision.

Côté API, ce REFUSEE EN CAL doit entraîner ce qui suit

1 – sur l'offre

RAS

2 – sur la demande

- Si la demande a été validée par l'entreprise, son entity_status est recalculé en tenant compte d'une éventuelle autre candidature présente sur la demande
- L'historique des statuts de la demande et le updated_at de la demande doivent être mis à jour.
- Les références à la candidature sont supprimées dans les champs : active_application_id ou dormant_application_ids, active_application_housing_offer_id, target_offer_reserving_business

3 – sur la candidature

- Le status passe à REFUSEE EN CAL. L'historique des statuts de la candidature et le updated_at de la candidature doivent être également mis à jour. Le code de l'événement, le motif de l'événement, la date de survenance de l'événement, la date de l'événement sont mis à jour.

Gestion de l'événement ANNRAD (annulation de radiation)

Suite à une erreur d'attribution par le bailleur, ce dernier envoie un événement ANNRAD sur la candidature pour propager la correction sur AL'in.

Côté API, ce TRANSMIS BAILLEUR doit entraîner ce qui suit

1 – sur l'offre

- Déclencher le passage de l'offre à offer_status = EN COURS D'ATTRIBUTION et lessor_status = EN ATTRIBUTION. L'historique des statuts de l'offre et le updated_at de l'offre doivent être également mis à jour.

2 – sur la demande

- Déclencher le passage de la demande à request_status = COMPLETE ou INCOMPLETE et entity_status = statut précédant « SALARIE LOGE ». L'historique des statuts de la demande et le updated_at de la demande doivent être également mis à jour.
- Si is_active=false il passe à true

- Les champs relatifs à l'offre attribuée au demandeur repassent à null : attributed_housing_district, attributed_housing_district_postal_code, attributed_housing_surface, lessor_name, lessor_siren.
- Les champs relatifs à la candidature en cours retrouvent des valeurs associées à la candidature réactivée : active_application_id ou dormant_application_ids, active_application_housing_offer_id, target_offer_reserving_business
- Le entity_status de la demande est recalculé.

3 - sur la candidature

- Le status passe à TRANSMIS BAILLEUR. L'historique des statuts de la candidature et le updated_at de la candidature doivent être également mis à jour. Le code de l'événement, le motif de l'événement, la date de survenance de l'événement, la date de l'événement et éventuellement le rang sont mis à jour.
- ~~Si le code motif est différent de EV08, il n'est pas enregistré.~~ **Changement du 18/06 : on enregistre tous les codes motif reçus**

A noter : si la réactivation de la candidature conduit à un nombre de candidatures actives ou dormantes non autorisé, une erreur sera renvoyée (à détailler dans la spécification technique).

Gestion de l'événement NPRCAL (non présentation en CAL)

Côté API, ce REFUSEE EN CAL doit entraîner ce qui suit

1 - sur l'offre

RAS

2 - sur la demande

- Si la demande a été validée par l'entreprise, son entity_status est recalculé en tenant compte d'une éventuelle autre candidature présente sur la demande
- L'historique des statuts de la demande et le updated_at de la demande doivent être mis à jour.
- Les références à la candidature sont supprimées dans les champs : active_application_id ou dormant_application_ids, active_application_housing_offer_id, target_offer_reserving_business

3 - sur la candidature

- Le status passe à REFUSEE EN CAL. L'historique des statuts de la candidature et le updated_at de la candidature doivent être également mis à jour. Le code de l'événement, le motif de l'événement, la date de survenance de l'événement, la date de l'événement sont mis à jour.
- ~~Si le code motif est différent de EV08, il n'est pas enregistré.~~ **Changement du 18/06 : on enregistre tous les codes motif reçus**

Gestion de l'événement DESCAL03 **(ajout du 18/06)**

Dans le cadre de cette feature, il est demandé qu'en cas de réception d'un **DESCAL03** avec le code motif **EV01**, alors nous n'ajoutons pas les 40 points (application_rejected_points) à la demande, contrairement à ce qui avait été demandé dans le [#21663](#) (besoin exprimé dans le devops 39367).

Pour ne pas avoir à faire un script de rattrapage, nous allons filtrer les candidatures qui ont soit le statut **REFUSEE EN CAL** si pas de code événement, soit celles qui ont un code événement **DESCAL03, DESCAL04 ou NPRCAL** et pas de code motif **EV01**.

Flux 5

Logique du flux 5

Les données sont envoyées à Al'in Gestion au format xml via l'appel développé par Sopra.

Changements du 19/06

Comme nous devons faire cohabiter 3 versions de flux 5 (v1, v2 et v3), il a finalement été décidé que l'on appellera une route différente en fonction de la version du flux 1 reçu. Les routes sopra sont (exemple sur l'env dem) :

<http://192.168.190.118:8080/SWSDEM/ws/rest/als/bien/v1/integrer> (version 1 du patrimoine)

<http://192.168.190.118:8080/SWSDEM/ws/rest/als/bien/v2/integrer> (version 2 du patrimoine)

<http://192.168.190.118:8080/SWSDEM/ws/rest/als/bien/v3/integrer> (version 3 du patrimoine)

Avant de faire l'appel nous ferons la vérification du bon format via un xsd (comme c'est déjà le cas actuellement). Cela permet de limiter les appels en erreur auprès de sopra. Cela signifie qu'il y a un xsd par appel, donc 3 xsd qui nous sont fournis par sopra.

Pour ce flux nous recréons un xml à partir des informations de l'offre enregistrées en base Al'in. Lors de la modification d'une offre (que ce soit via flux ou front), s'il existe une différence dans le xml généré, alors l'envoi vers Al'in Gestion est déclenché.

La logique technique est la suivante :

- Réception d'un appel de création ou de mise à jour d'offre (depuis Ikos Wrapper ou front bailleur)
- On vérifie dans le callback après sauvegarde de l'offre si elle est mise à disposition (is_mad = true) et si un changement a été fait sur n'importe quel champ de l'offre
- Un job est créé en tâche de fond, chargé de générer le xml à envoyer dans le cadre de ce flux
- Si le xml généré est différent du xml précédemment envoyé (on utilise les valeurs précédentes sur l'offre), alors l'appel "integrer" est fait vers Al'in Gestion (la liste des champs présents dans le xml est indiquée en annexe, Flux 5 unifié)
- Si le xml généré ne correspond pas au xsd associé à la version du flux 1, alors nous créons une entrée dans la table **SopraXmlSender** (champs last_xml_validation_report, last_send_has_error, xml_sopra_error_ids, hash_xml, last_successful_xml_sent_date) et
- Si l'appel réussi, alors on met à jour sur l'offre :
 - o hash_xml : avec le xml envoyé dans
 - o updated_at : Avec l'heure actuelle
 - o last_successful_xml_sent_date : avec la date du jour

Nous ne créons pas de log en cas d'échec de l'appel.

Flux 5 unifié

Concernant la création du XML, il a été décidé que le même xml soit envoyé à Sopra. Dans ce flux 5 unifié, les valeurs transmises changeront en fonction de ce qu'il y a en base de données, mais le contrôle des champs obligatoires et formats attendus sera fait par Al'in Gestion lors de la réception en fonction du numéro de version.

Changements pour la création du xml

Il faut adapter le mapping de valeur nécessaire à la recreation du xml pour exclure les champs supprimés, ajoutés les champs créés.

Il faut également prévoir un traitement particulier pour décomposer au besoin la valeur de parking_type pour enregistrer chaque élément dans une balise stationnement_logement_nature distincte, sous une balise stationnement_logement, lors de l'envoi du flux 5.

Nous devons également ajouter une balise "saisie_alin" correspondant au champ api "is_from_form".

Référentiel de typologies de logement

Actuellement les valeurs possibles du champ `typology` sur le modèle `housing_offers` sont **C, T1, T2, T3, T4, T5, T6P**. Cette liste est remplacée par : C, T1, T2, T3, T4, T5, **T6, T7, T8, T9P**. Cependant même si la valeur T6P n'est plus attendue, elle sera acceptée pour les offres existantes. Les nouvelles valeurs **T6, T7, T8, T9P** seront à traiter de façon identique à **T6** et **T6P** en termes d'éligibilité. Il existe également des valeurs **T1P** et **T1B**, mais elles peuvent être ignorées dans le cadre du logement social.

Il faut désormais accepter les nouvelles valeurs **T6, T7, T8, T9 et T9P** et les traiter comme du T6P en termes d'éligibilité. C'est-à-dire qu'une personne précédemment éligible à du T6P sera désormais éligible aux T6, T7, T8, T9, T9P.

Modification de l'ihm collaborateur ALS

Nous utilisons actuellement le champ `priority_public` présent sur l'offre pour afficher l'information dans le champ "Public prioritaire" sur la page de détail d'une offre (exemple en recette 2) :

■ Accueil / Retour

Offre 61dbfdb8b93ee31090907eaf

- Détails de l'offre -

Date de création 10/01/2022 10:34:45 Taux d'effort min. 0.333 Commune Besné Pays France Montant du loyer avec charges 370 Détails Orientations Droit commun Score obtenu 1804 Points mobilité 0 Points - de 30 ans 0 Non retenu repris bailleur 0 Lien sur la commune du lieu de travail 0	Statut de l'offre LOGEMENT PLACE Typologie Département 44 Adresse 8 RUE DE L'ECOLE Surface 40 Parent isolé Points validation entreprise 1000 Points rapprochement du lieu de travail 0 Non retenu en CAL 0 Handicap 0 Equilibrage scoring 0	État du logement chez le bailleur PLACE Type Code postal 44180 Montant du loyer 320 Description Public prioritaire Non Points entreprise 0 Points ancienneté demande/SNE 804 Non retenu au moment de la transmission bailleur 0 Victime de violences familiales 0
--	---	--

- Scoring de l'applicant -

Nous devons modifier l'ihm collaborateur pour ne plus utiliser ce champ mais vérifier la présence de `priority_public` dans le champ **housing_channel_keys**.

Nous devons également supprimer la déclaration des champs **first_quartile** et **other_quartile** dans une optique de nettoyage. Ceux-ci ne sont pas utilisés au niveau de l'interface de l'ihm.

Ajout de tests pour le flux 5

Tests :

- Bon fonctionnement du flux 5 (déclenchement de l'envoi, génération du xml, envois et enregistrement d'envoi réussi)
- Gestion du non envoi lors que le xml généré est identique au dernier envoi
- Validation du xml par le xsd
- Gestion des erreurs au différent niveau du processus

Script de traitement de l'existant

Pour toutes les demandes qui ont dans `typology_range` T6, T6P, T7, T8, T9 ou T9P, on recalcule l'éligibilité pour contenir T9 et T9P en plus.

Les champs à loguer en sortie seront :

- `Housing_request_id`
- `Als_id`
- `Unique_numbers`
- `Housing_request.typology_range`
- `Housing_request.eligibility_query_base_before`
- `Housing_request.eligibility_query_base_after`
- `Housing_request.old_updated_at`

ANNEXES

Flux 5 unifié

Les champs indiqués en rouges correspondent aux champs api. Ce sont donc ceux sur lesquels une modification entraîne l'envoi du flux 5 vers Al'in Gestion :

```
<ns2:patrimoine xmlns:ns2="http://v1.als.ikos.webservice.immobilier.sopra/patrimoineALS">
  <informations_flux>
    <version_flux>[dataflow_version]</version_flux> (ajouté en v2 et v3)
    <type_flux>BIEN</type_flux>
    <date_flux>[date du jour]</date_flux>
  </informations_flux>
  <bien>
    <SIREN_bailleur>[lessor_siren]</SIREN_bailleur>
    <saisie_alin>[is_from_form -> 0/N]</saisie_alin> ajout utile pour les repris bailleur, etc
    <libelle_organisme>[lessor_name]</libelle_organisme>
    <code_territoire>[lessor_code]</code_territoire>
    <SIREN_ALS>[als_siren]</SIREN_ALS>
    <reference_offre>[id]</reference_offre>
    <reference_MAD>[reference_mad]</reference_MAD>
    <date_MAD_ALS>[mad_als_date]</date_MAD_ALS>
    <id_rpls>[rpls_id]</id_rpls>
    <date_disponibilite>[availability_date]</date_disponibilite>
    <date_livraison>[delivery_date]</date_livraison> (spécifique v3, facultatif)
    <date_derniere_rehabilitation>[latest_rehabilitation]</date_derniere_rehabilitation>
    <date_fin_designation>[designation_date_end]</date_fin_designation>
    <orientations_bailleur>
      <orientation>[aucune info à fournir à ce stade]</orientation>
    </orientations_bailleur>
    <reference>[external_ref]</reference>
    <texte>[description]</texte>
    <adresse>[address]</adresse>
    <adresse_complement>[address_complements]</adresse_complement>
    <adresse_complement_2>[address_extended]</adresse_complement_2> (spécifique v3, facultatif)
    <numero_porte>[door_number]</numero_porte> (spécifique v3, facultatif)
    <communiquer_adresse_exacte>[address_can_be_shared -> 0/N]</communiquer_adresse_exacte>
    <code_postal>[postal_code]</code_postal>
```

```

<ville_insee>[insee_code]</ville_insee> (spécifique v3, obligatoire en v3)
<departement>[department]</departement>
<ville>[district]</ville>
<code_residence>[residence_code]</code_residence>
<residence_type>[residence_type]</code_residence> (spécifique v3, obligatoire en v3)
<libelle_residence>[residence_title]</libelle_residence>
<regles_vie_residence>[residence_rules]</regles_vie_residence> (spécifique v3, facultatif)
<loggia>[loggia]</loggia> (spécifique v3, facultatif)
<cellier>[cellar]</cellier> (spécifique v3, facultatif)
<type_cuisine>[kitchen_type]</type_cuisine>
<presence_gardien>[has_gardien]</presence_gardien>
<nature_stationnement>[parking_type]</regles_vie_residence> (spécifique v2, facultatif)
<stationnements_logement> (spécifique v3, facultatif)
  <stationnement_logement_nature>[parking_type]</stationnement_logement_nature> (spécifique v3,
facultatif)
</stationnements_logement> (spécifique v3, facultatif)
<loyer_mensuel>[rent_amount]</loyer_mensuel>
<loyer_mensuel_maximum>[monthly_rent_max]</loyer_mensuel_maximum> (spécifique v3, facultatif)
<charges_locatives>[rental_charges]</charges_locatives>
<loyer_annexe_charges_comprises>[annex_rent_with_charges]</loyer_annexe_charges_comprises>
(spécifique v3, facultatif)
  <loyer_mensuel_cc>[rent_with_charges]</loyer_mensuel_cc> (spécifique v2, obligatoire) ATTENTION
CETTE BALISE SERA AUSSI ENVOYEE EN V3 (avec une valeur calculée utilisée par l'API), IL FAUDRA BIEN
S'ASSURER DE L'IGNORER COTE AL'IN GESTION POUR UNE OFFRE EN V3
  <montant_charges_externes>[external_charges]</montant_charges_externes> (spécifique v3,
facultatif)
<depot_garantie>[guarantee_deposit]</depot_garantie>
<loyer_stationnement_cc>[parking_charges]</loyer_stationnement_cc>
<loyer_stationnement_cc_obligatoire>[parking_charges_mandatory]</loyer_stationnement_cc_
obligatoire> (spécifique v3, facultatif)
<meuble>[furnished_lodging]</meuble>
<ref_convention_reservation>[construction_type]</ref_convention_reservation>
<raison_sociale_reservataire>[reserving_business_name]</raison_sociale_reservataire>
<SIREN_reservataire>[reserving_siren]</SIREN_reservataire>
<id_externe_pdv>[agency_id]</id_externe_pdv>
<libelle_agence>[agency_name]</libelle_agence>
<contact_a_afficher>[lessor_contact_name]</contact_a_afficher> (spécifique v2, facultatif)
<email_a_afficher>[lessor_email]</email_a_afficher> (spécifique v2, facultatif)
<telephone_a_afficher>[lessor_phone_number]</telephone_a_afficher> (spécifique v2, facultatif)
<contact_commercial_nom>[commercial_contact{lastname}]</contact_commercial_nom> (spécifique v3,
obligatoire)

```

```

    <contact_commercial_prenom>[commercial_contact{firstname}]</contact_commercial_prenom>
    (spécifique v3, facultatif)

    <contact_commercial_email>[commercial_contact{email}]</contact_commercial_email> (spécifique v3,
    obligatoire)

    <contact_commercial_telephone>[commercial_contact{phone_number}]</contact_commercial_
    telephone> (spécifique v3, facultatif)

    <contact_candidature_nom>[application_manager{lastname}]</contact_candidature_nom> (spécifique
    v3, obligatoire)

    <contact_candidature_prenom>[application_manager{firstname}]</contact_candidature_prenom>
    (spécifique v3, facultatif)

    <contact_candidature_email>[application_manager{email}]</contact_candidature_email> (spécifique
    v3, obligatoire)

    <contact_candidature_telephone>[application_manager{phone_number}]</contact_candidature_
    telephone> (spécifique v3, facultatif)

    <contacts> (spécifique v3, facultatif)
        <contact> (spécifique v3, facultatif)
            <contact_fonction>[contacts[0]{position}]</contact_fonction> (spécifique v3, facultatif)
            <contact_nom>[contacts[0]{lastname}]</contact_nom> (spécifique v3, facultatif)
            <contact_prenom>[contacts[0]{firstname}]</contact_prenom> (spécifique v3, facultatif)
            <contact_telephone>[contacts[0]{phone_number}]</contact_telephone> (spécifique v3,
            facultatif)
        </contact>
    </contacts>

</bien>
</ns2:patrimoine>

```